

## 2.5 トランスポート層

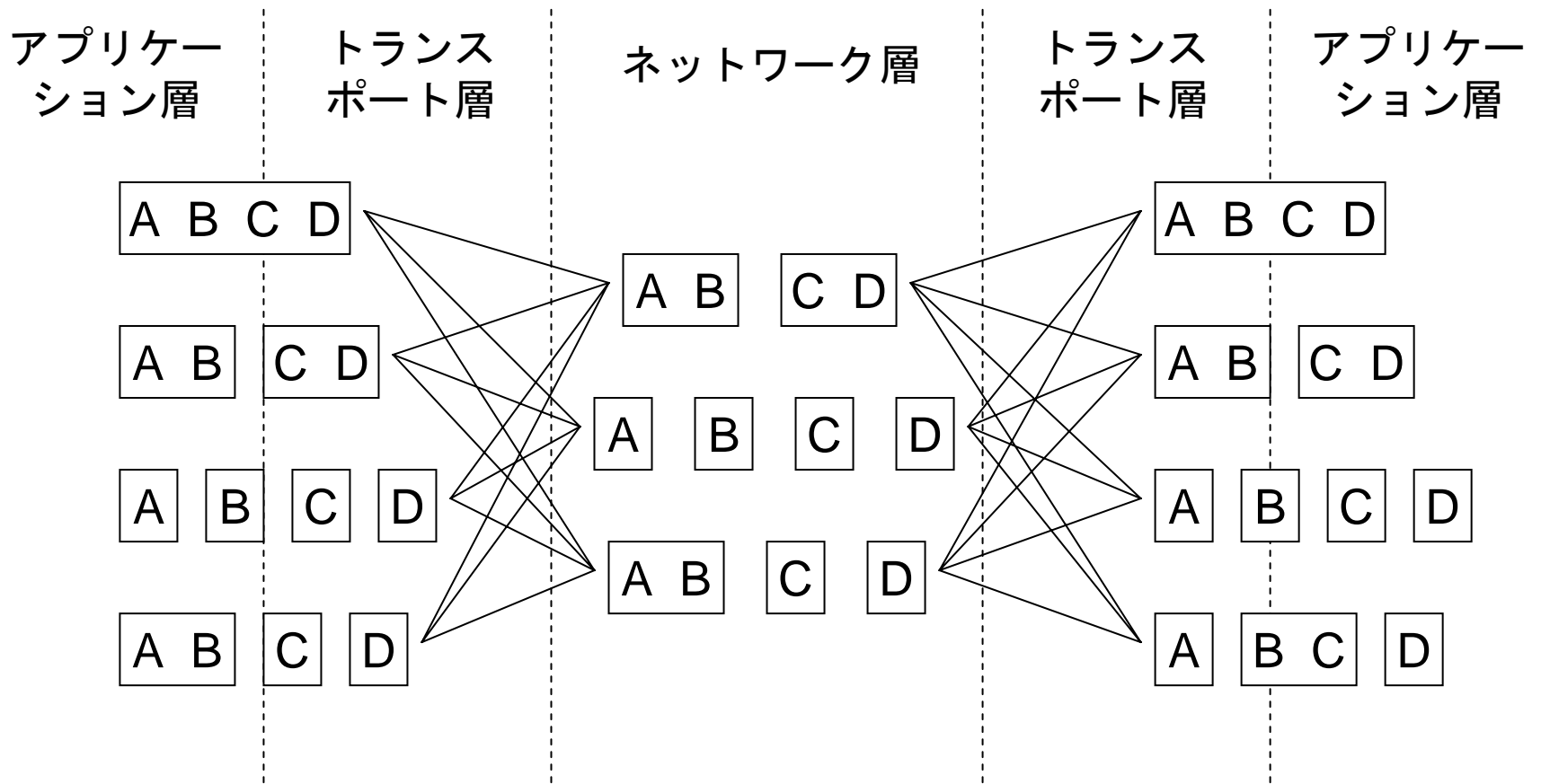
# TCPとUDP

- TCP (Transmission Control Protocol)
    - コネクション型、ギャランティード
    - マルチキャスト・ブロードキャスト不可
  - UDP (User Datagram Protocol)
    - コネクションレス、ベストエフォート
    - マルチキャスト・ブロードキャスト可
- cf. IP (Internet Protocol)
- コネクションレス、ベストエフォート

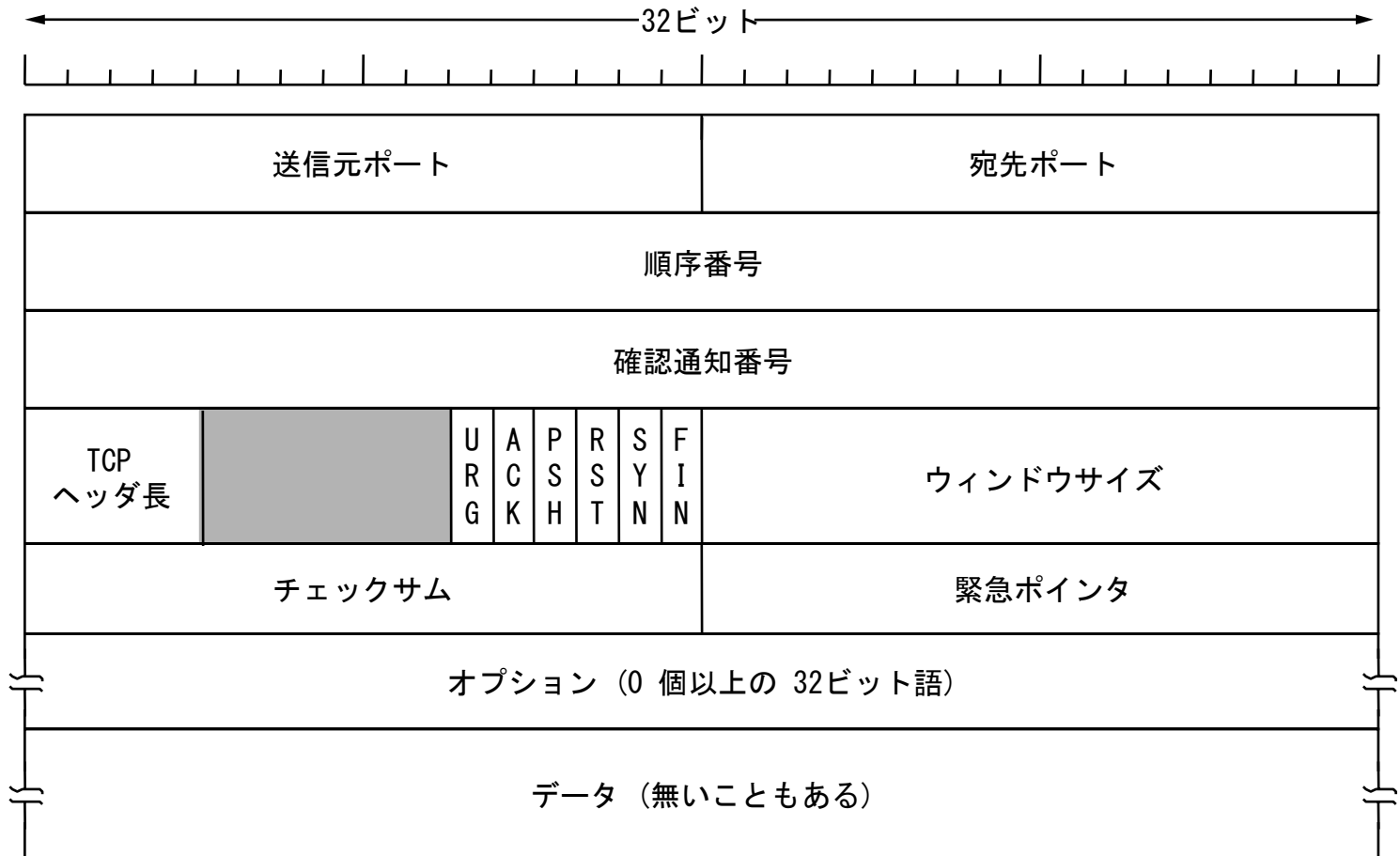
# ポート番号の例

ポート番号	プロトコル名	アプリケーション
21	FTP	ファイル転送
23	telnet	遠隔ログイン
25	SMTP	電子メール
53	DNS	ドメイン名
80	HTTP	WWW
110	POP-3	電子メール読み出し
143	IMAP	電子メール読み出し
443	HTTPS	セキュアWWW

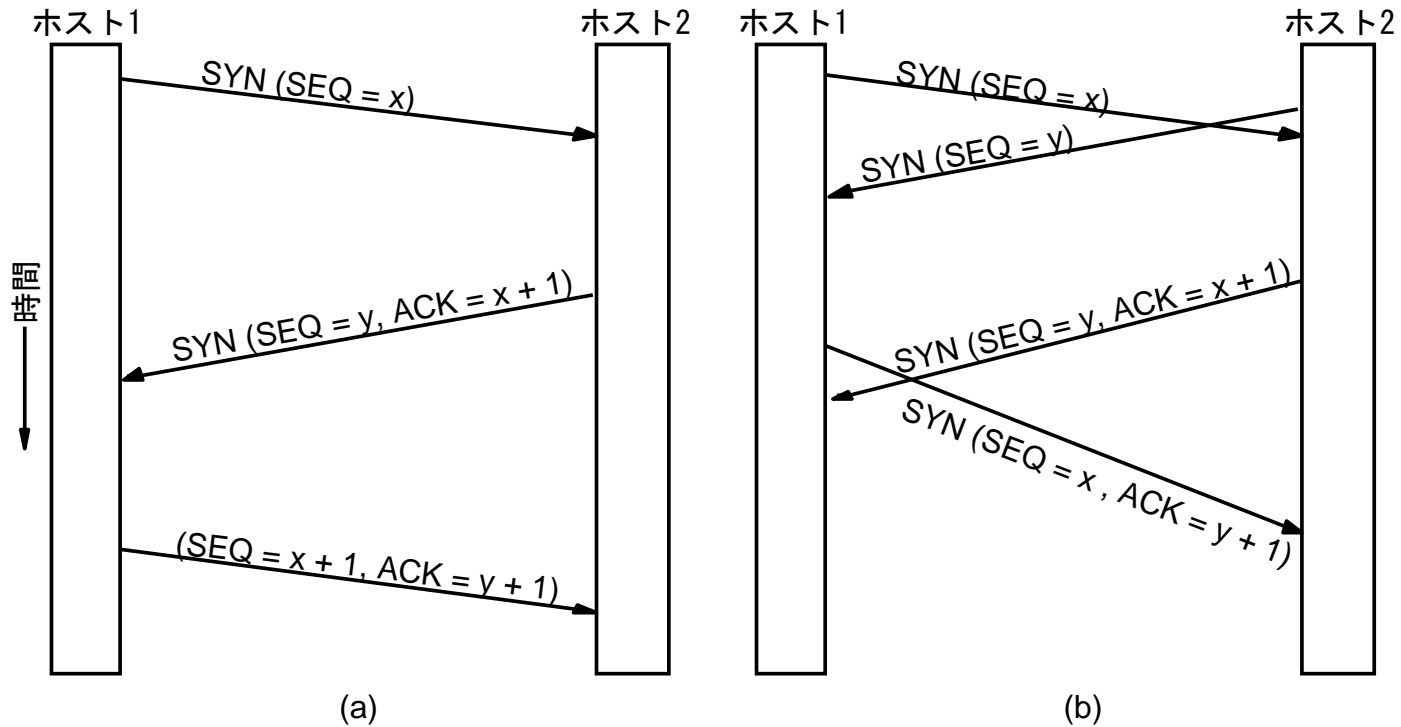
# TCPのサービスモデル



# TCPセグメント



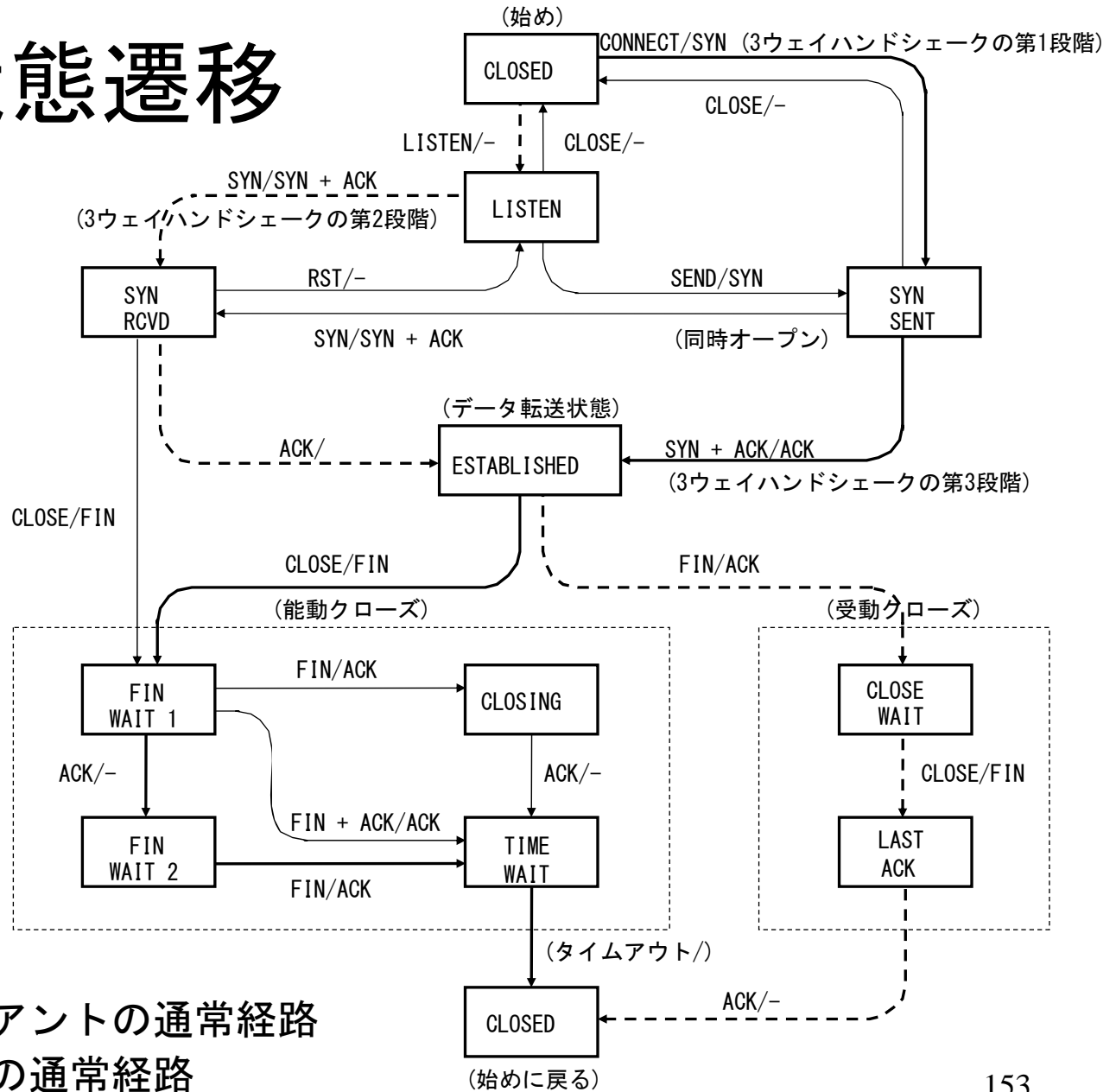
# TCPのコネクション確立



(a) 通常のTCPコネクション確立

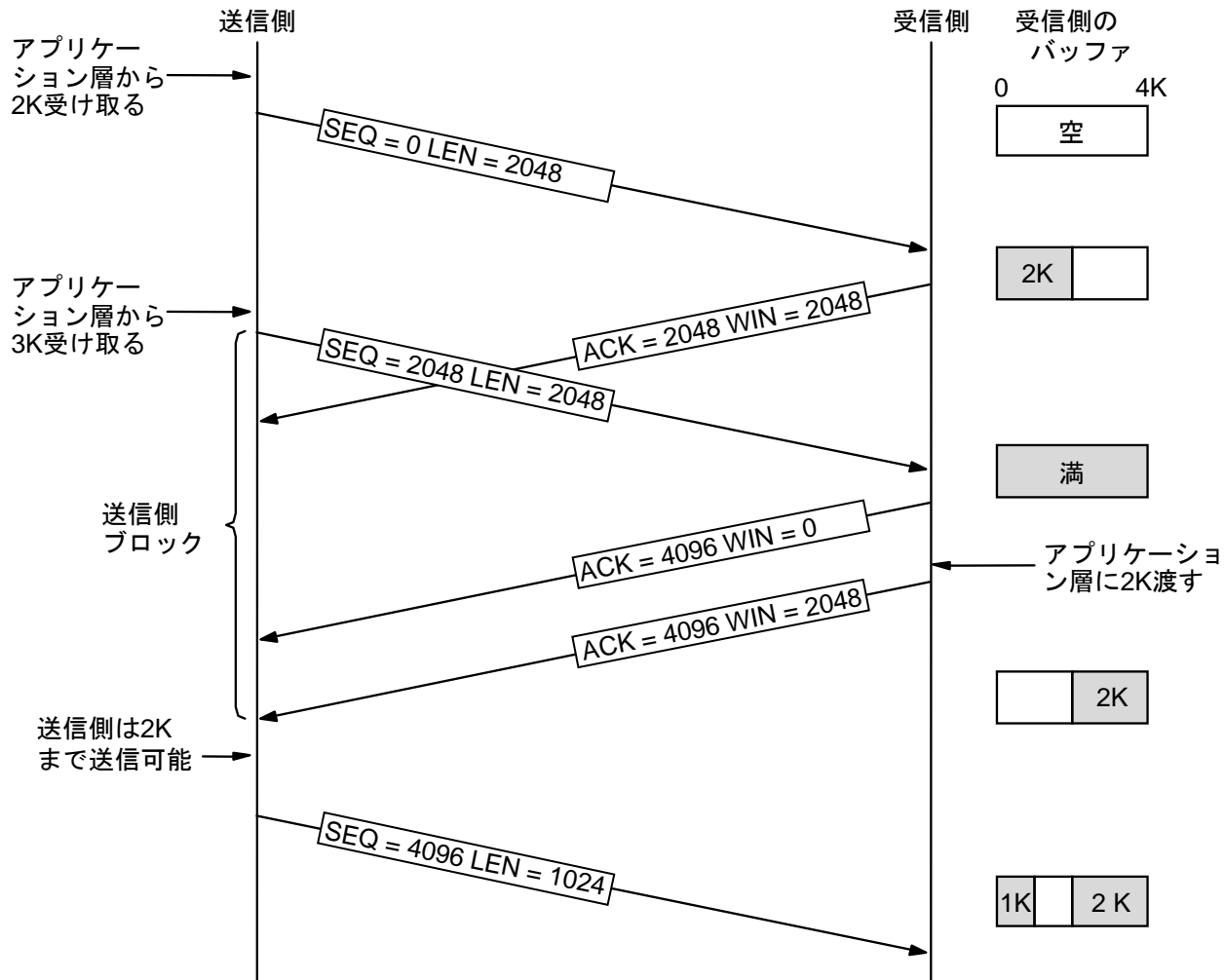
(b) 呼び出しの衝突

# TCPの状態遷移



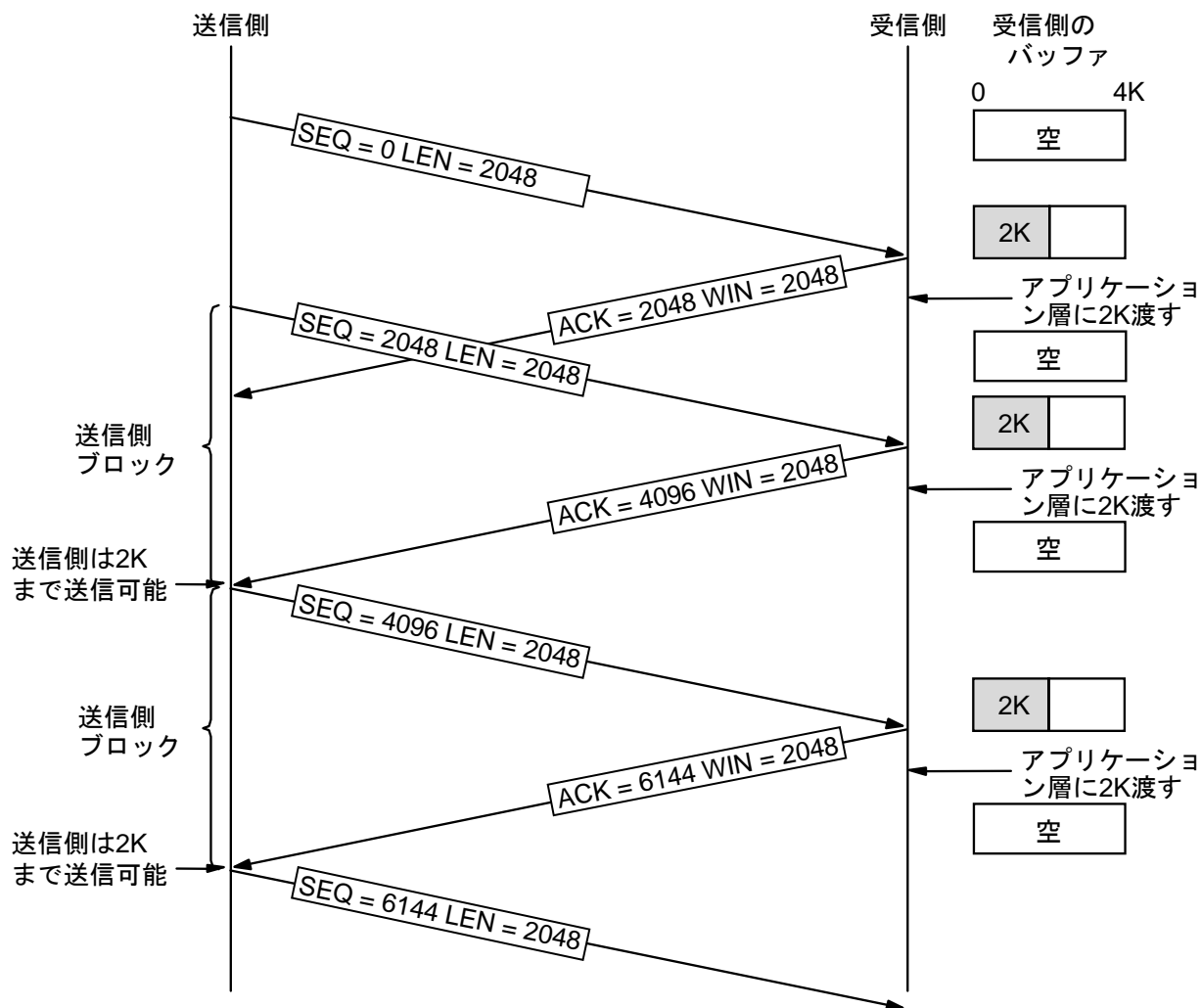
太い実線 : クライアントの通常経路  
 太い点線 : サーバの通常経路

# TCPのフロー制御





# アプリケーション層が高速な場合



# 連続送信可能ウィンドウサイズ

- ACKが戻ってくるよりウィンドウサイズが大きければ連続的に送信可能

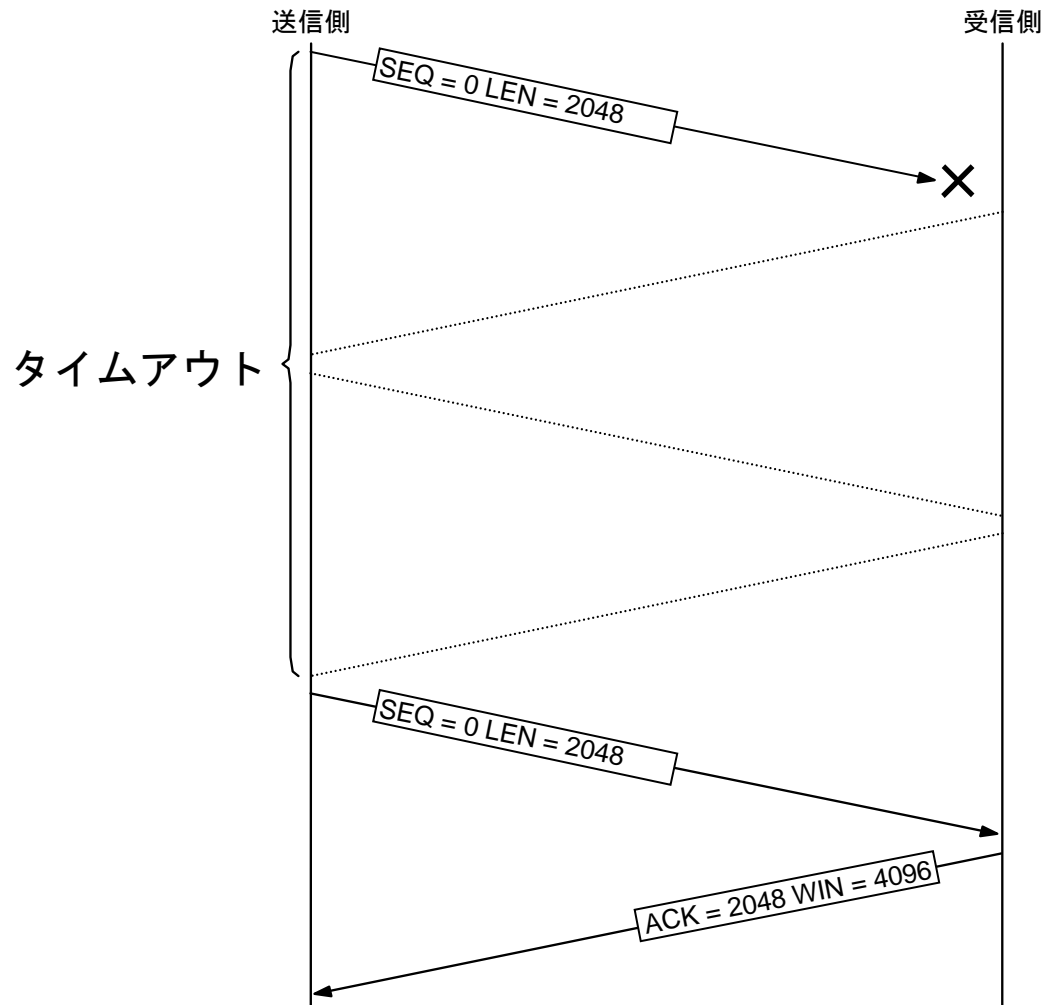
例：100Mbps, 片道20km  $\div$  0.1ms

→ウィンドウサイズ: 20kbit=2500バイト

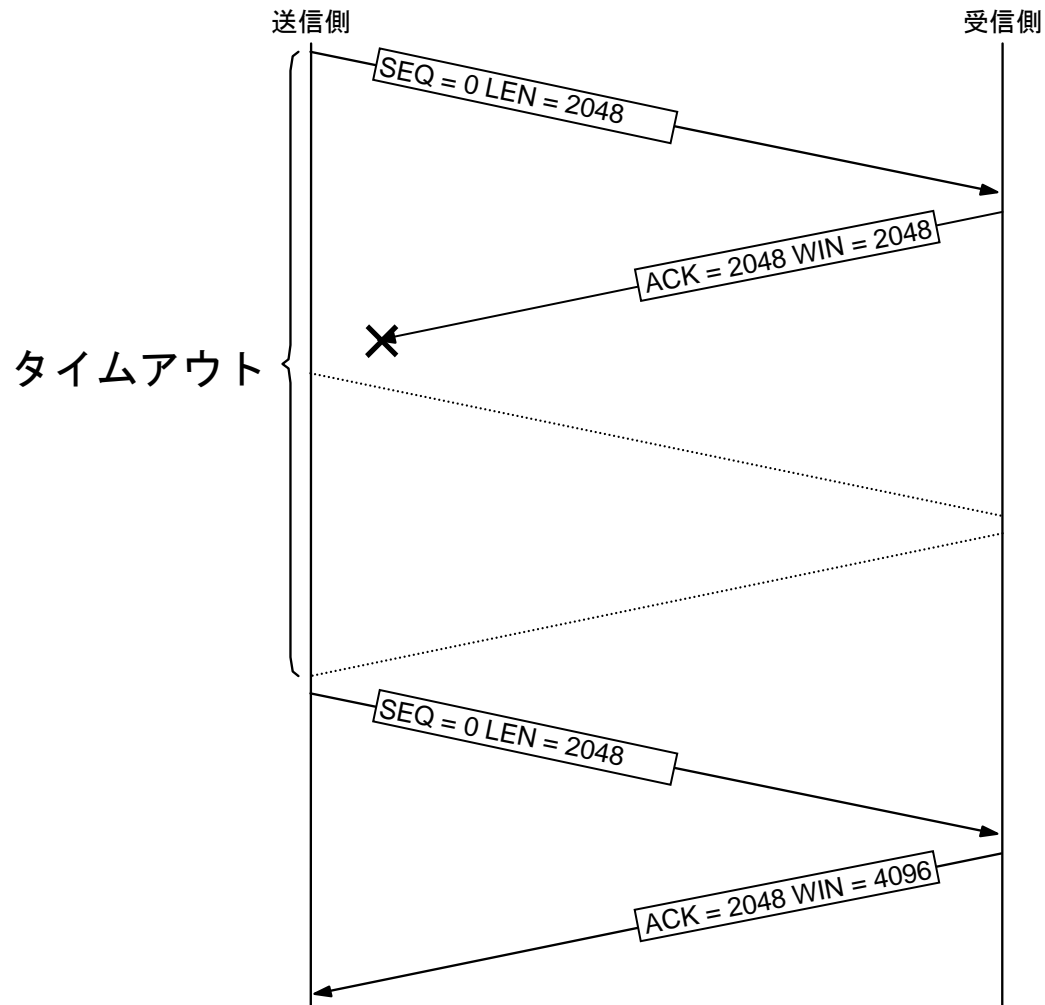
例：2Mbps, 往復200ms

→ウィンドウサイズ: 400kbit=50,000バイト

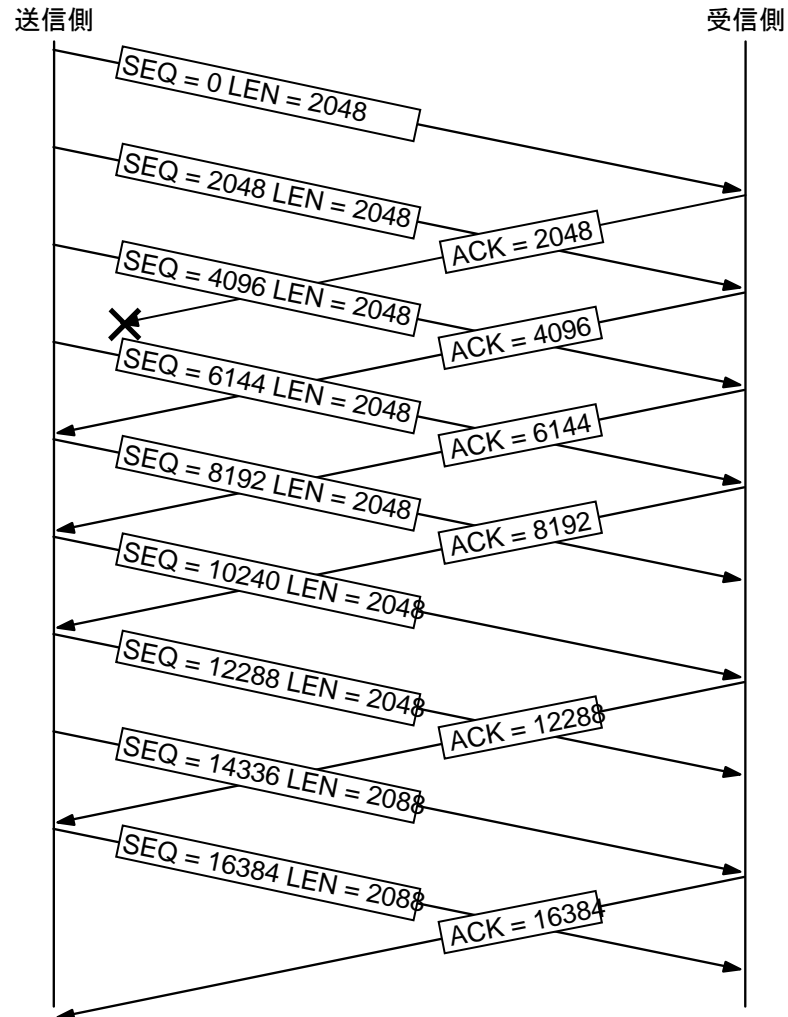
# TCPの誤り制御



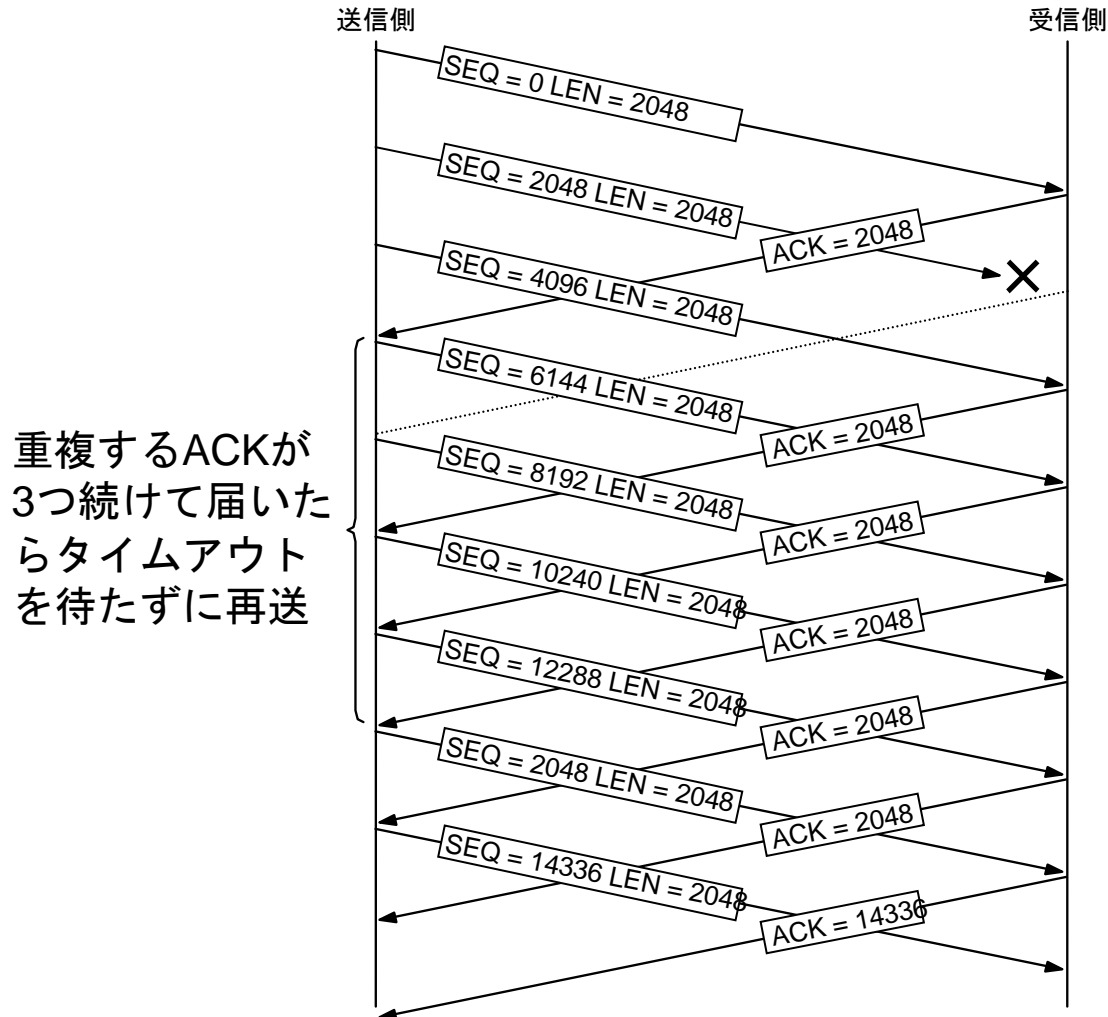
# ACKが失われた場合



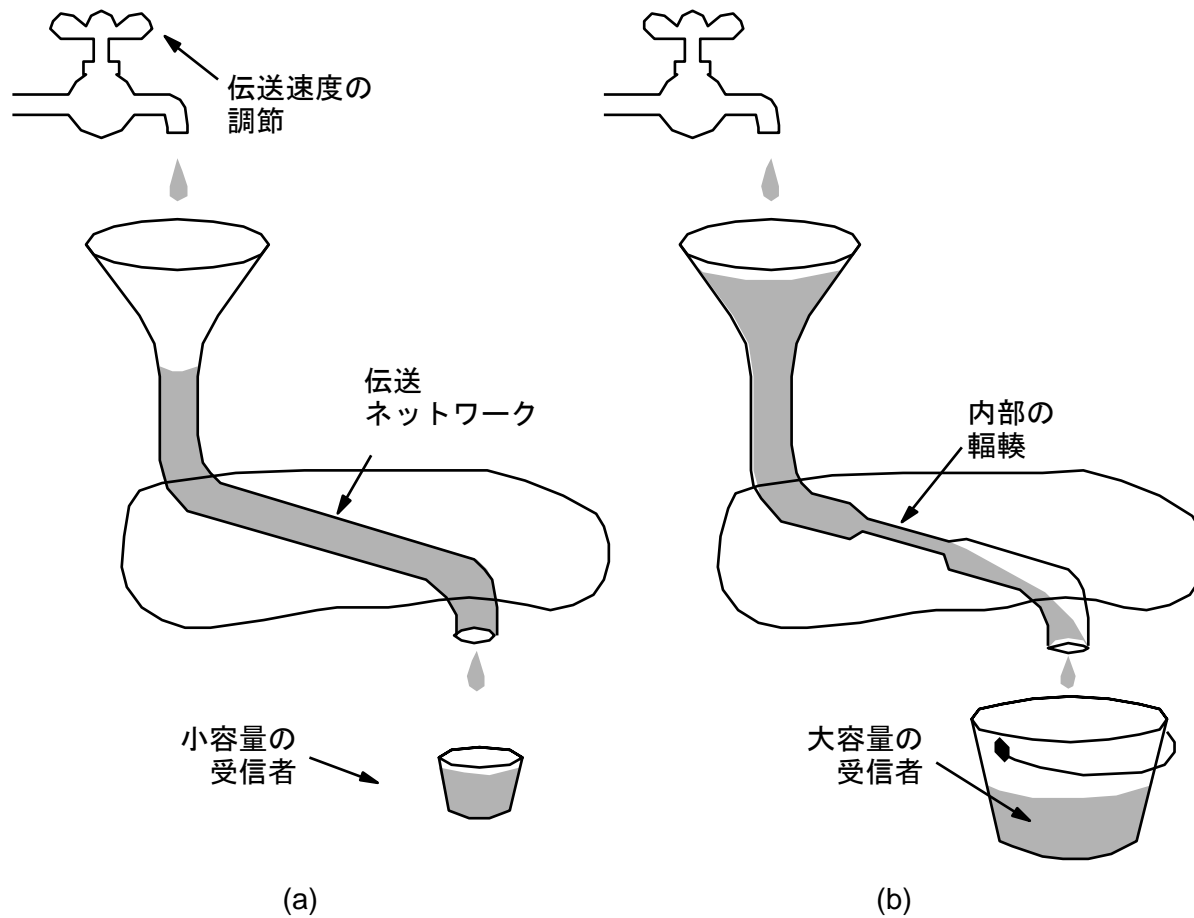
# 累積ACK



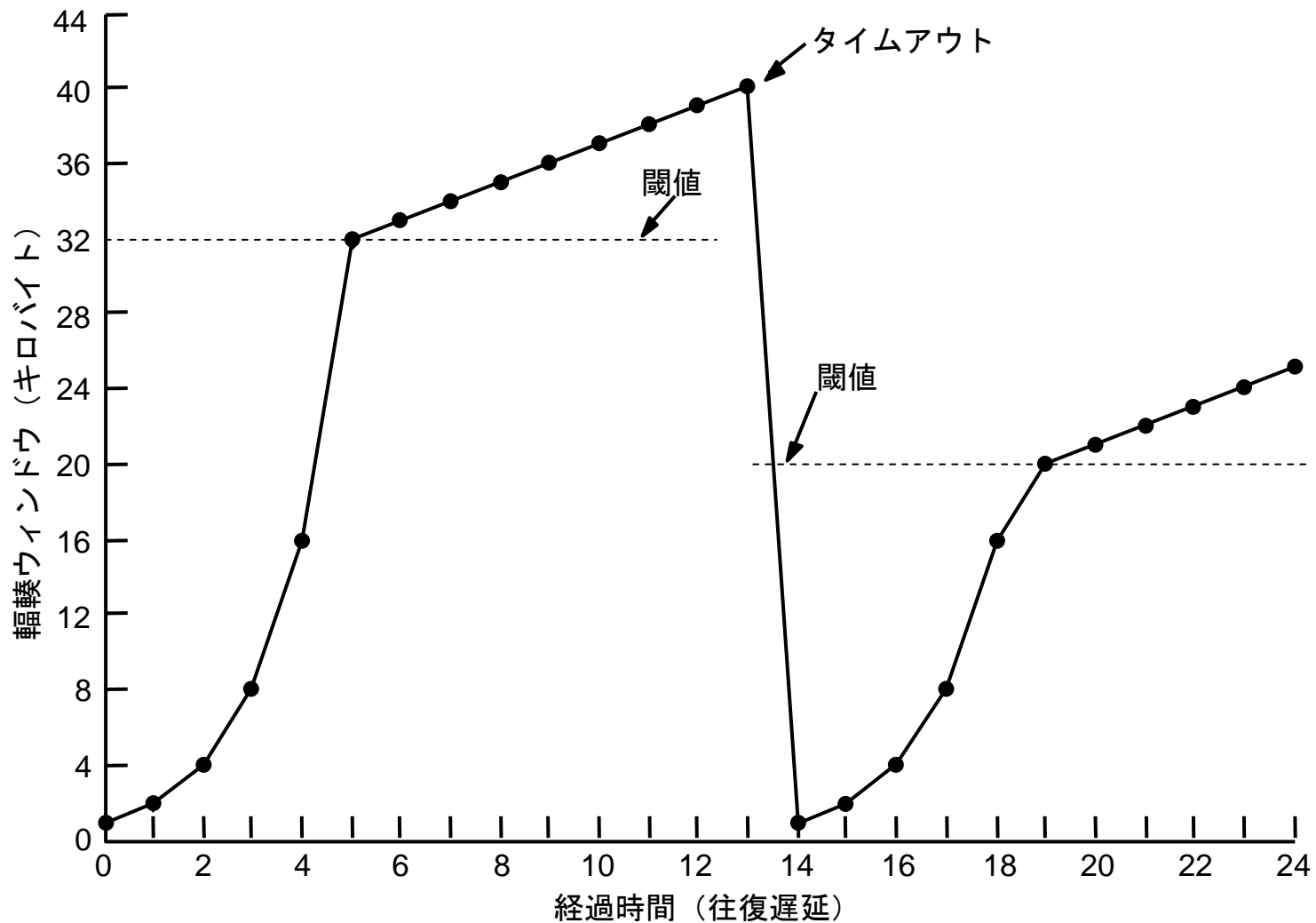
# Fast Retransmit



# フロー制御と輻輳制御



# スロースタート・輻輳回避

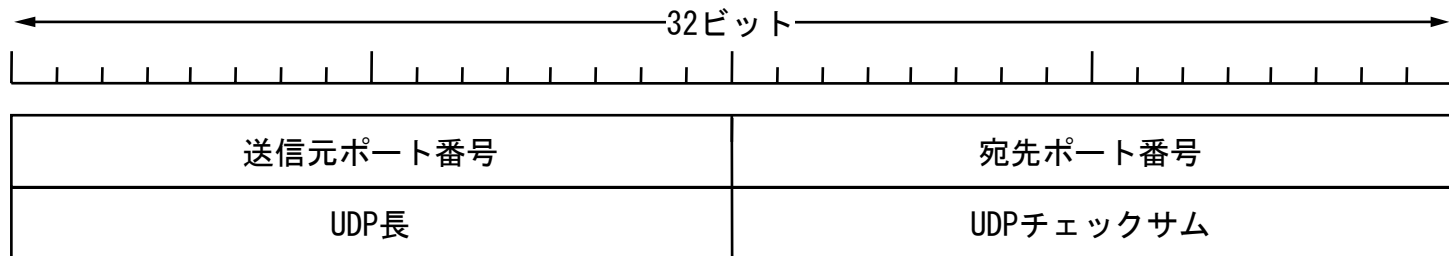




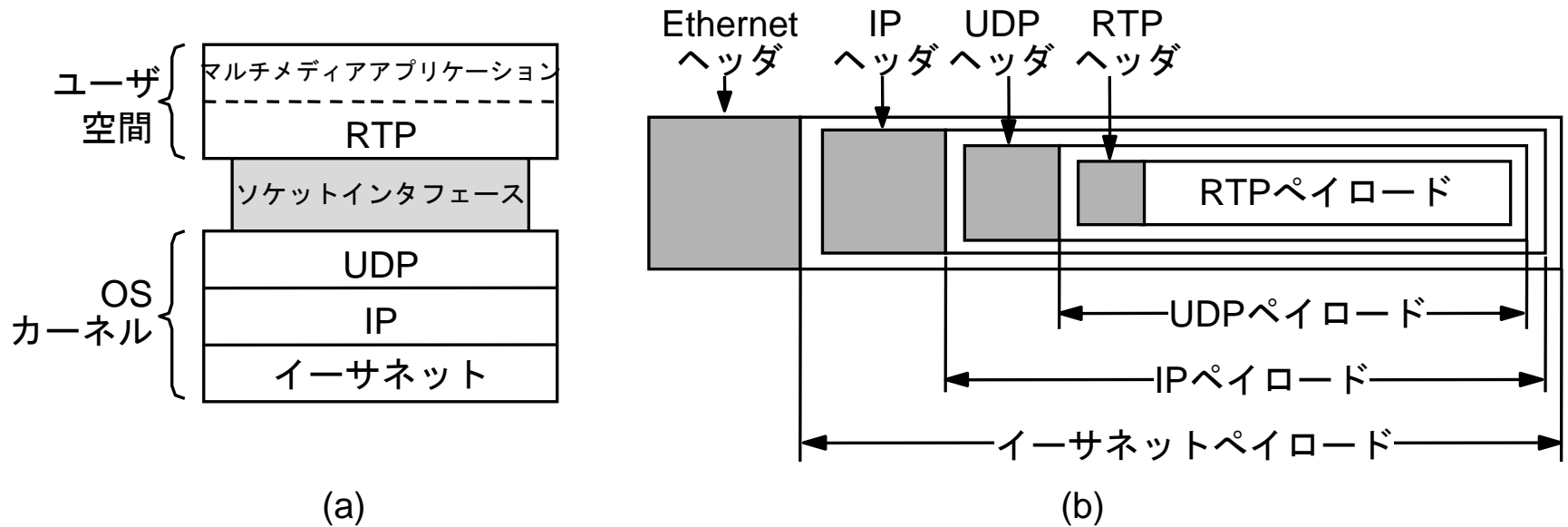
# 輻輳（パケットロス）の検出

- タイムアウト
  - 閾値 ← 未確認データ量  $\times 1/2$
  - 輻輳ウィンドウ ←  $1MSS$
- 3個の重複するACK
  - 閾値 ← 未確認データ量  $\times 1/2$
  - 輻輳ウィンドウ ← 閾値

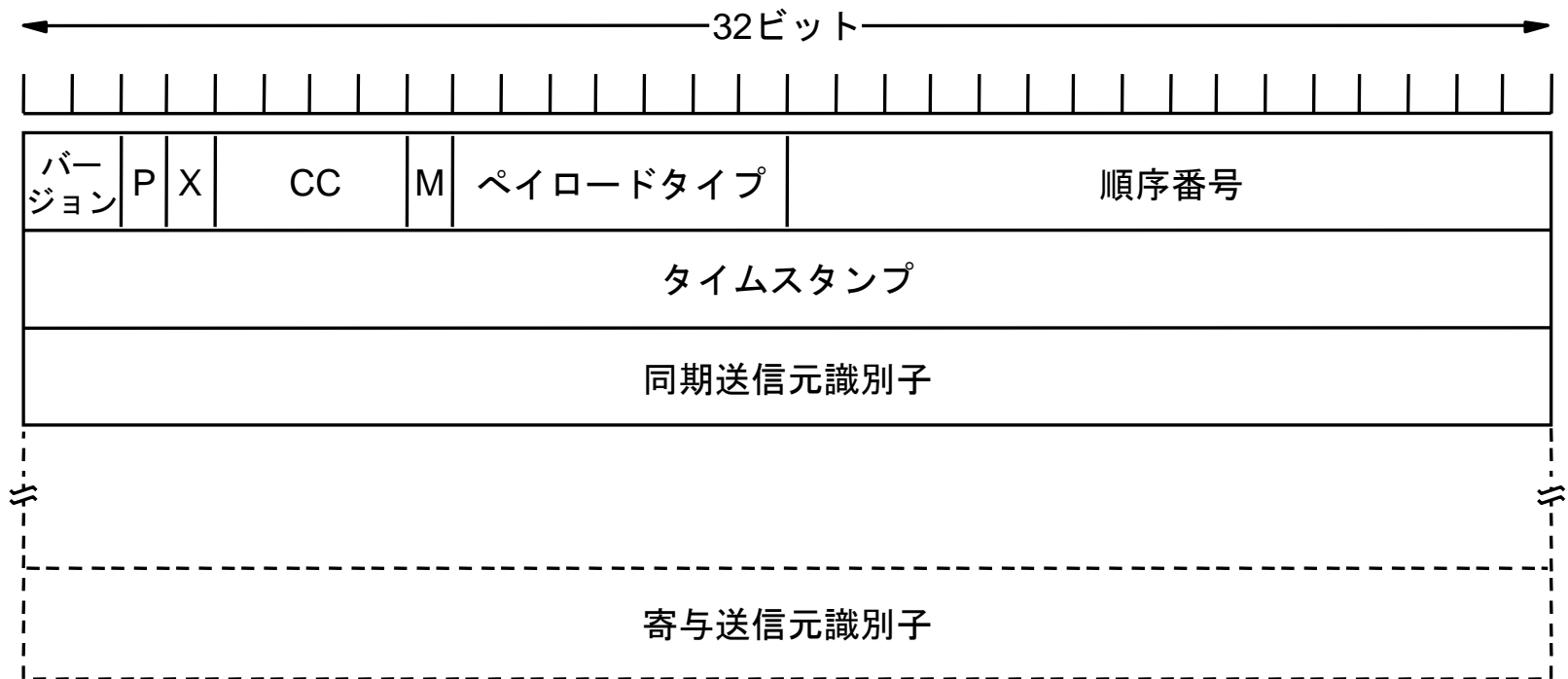
# UDPヘッダ



# RTP (Real-Time Transport Protocol)

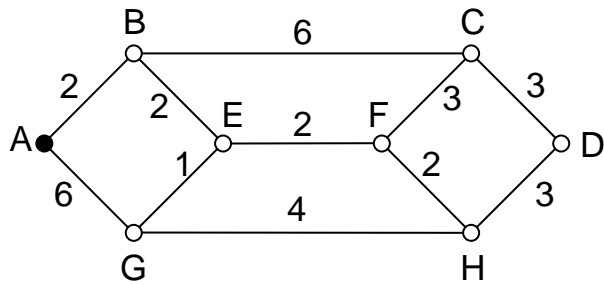


# RTPヘッダ

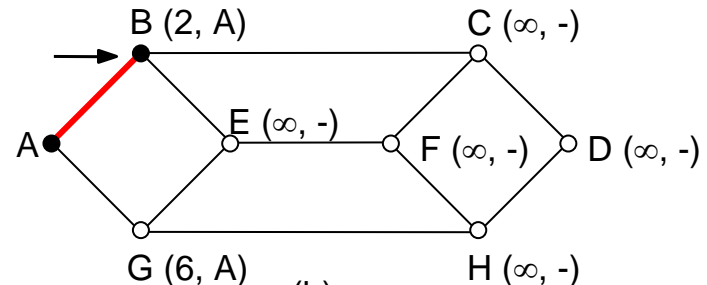


## 2.6 経路制御

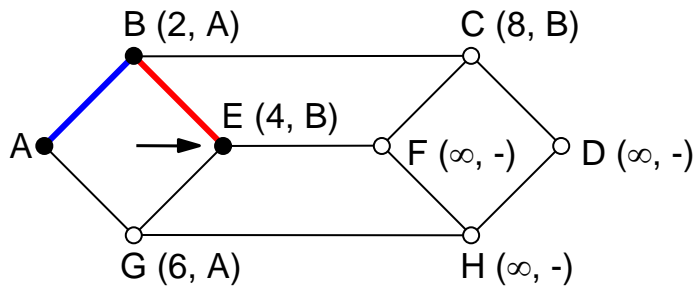
# Dijkstraのアルゴリズム



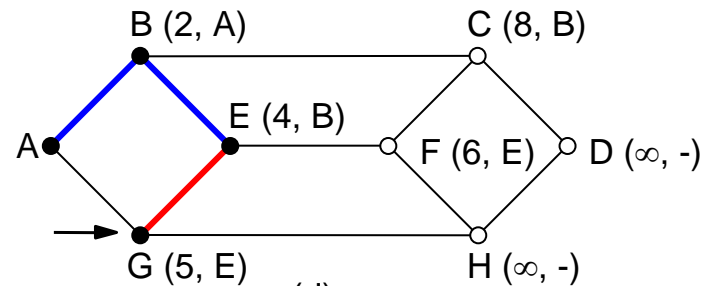
(a)



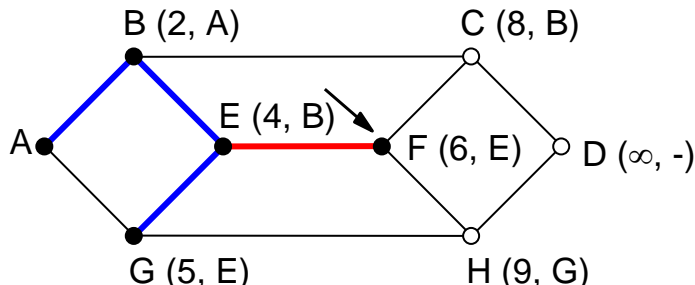
(b)



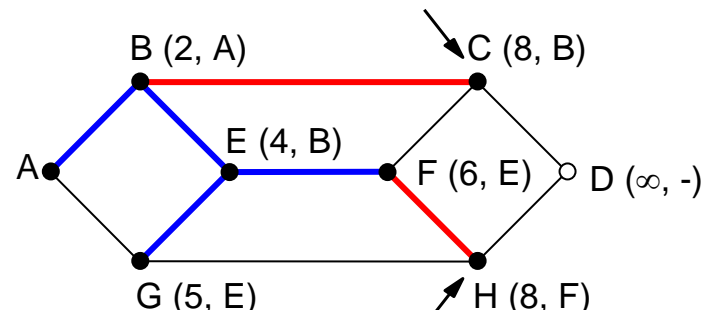
(c)



(d)



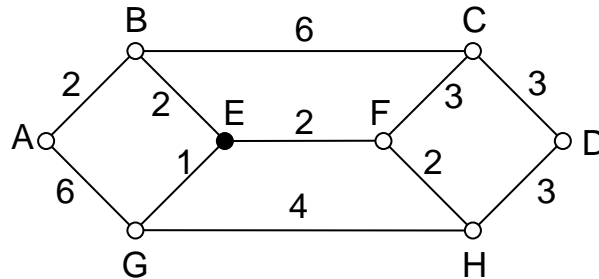
(e)



(f)

# 距離ベクトル方式 (Bellman-Ford のアルゴリズム)

- 宛先ノードまでの遅延の見積値を隣接ノード間で交換
- 受け取った見積値にそのノードまでの遅延を加えた値を比較し、最小のものを選択



- Bから Fから Gから 計算結果

A	2
B	0
C	6
D	9
E	2
F	4
G	3
H	6

A	6
B	4
C	3
D	5
E	2
F	0
G	3
H	2

A	5
B	3
C	6
D	7
E	1
F	3
G	0
H	4

A	4	B
B	2	B
C	5	F
D	7	F
E	0	-
F	2	F
G	1	G
H	4	F



# link state方式

- 隣接ノードとそこに至る遅延時間を他のすべてのノードに伝える  
→各ノードにおいてDijkstraのアルゴリズムにより最短経路を計算

# 経路制御プロトコルの例

- RIP (Routing Information Protocol)
  - 距離ベクトル
  - version 2でサブネットに対応
- OSPF (Open Shortest Path First)
  - link state
- BGP (Border Gateway Protocol)
  - 自律システム(≒プロバイダ)間の経路制御
  - 距離ベクトルだが途中経路をすべて伝える